This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (currently amended): A system for concurrent operation of plural computer applications, said system comprising:

(a) a computer storage medium including a shared object space selectively connectable to each of a plurality of computer applications, said shared object space capable of storing:

(i) a plurality of objects accessible to each of said plural computer applications connected to said shared object space; and

(ii) a queue associated with said shared object space and capable of storing a plurality of references to objects, a reference to an object received from ~~at least~~ one of said plural computer applications and identifying an individual object and ~~an~~ said application placing said reference to said individual object in said queue; and

(b) at least one computer comprising at least two computer applications concurrently executing on respective virtual machines, a particular object updateable by a first of said concurrently executing applications when said ~~one~~ first application is connected to said shared object space and ~~in control of~~ a datum in said object identifies a reference from said queue, said reference identifying said ~~particular object~~ first application, said first application relinquishing control of said ~~reference to said~~ particular object ~~to enable~~ enabling updating of said particular object by another application ~~when said reference to~~ with replacement of said datum in said particular object with another datum that identifies another ~~application as an application placing said~~ reference in said queue, said another reference identifying said another application, said first application and said another application not communicating with each other regarding said reference and said another reference.

Claim 2 (original): The system of claim 1 where said queue is a predefined type.

Claim 3 (original): The system of claim 1 where said queue is customized.

Claim 4 (original): The system of claim 1 where said queue is a "first-in-first-out" queue.

Claim 5 (original): The system of claim 1 where said queue is a "last-in-first-out" queue.

Claim 6 (previously presented): The system of claim 1 where each said virtual machine comprises a computer executable instruction in conformance with a virtual machine specification, said instruction executing on said computer.

Claim 7 (previously presented): The system of claim 6 where said shared object space is connected to each said virtual machine through a Native Method Interface.

Claim 8 (original): The system of claim 7 where said system includes a default directory with a native language library file.

Claim 9 (original): The system of claim 1 where said shared object space is operably connectable to a non-object-oriented application.

Claim 10 (previously presented): The system of claim 9 where said non-object oriented application is a "C" program.

Claim 11 (previously presented): The system of claim 1 where access to at least one of said plurality of objects by said plural computer applications is synchronized.

Claim 12 (previously presented): The system of claim 1 where said reference to an object further comprises an indication of a position of said reference in a sequence of references to objects placed in said queue by an application.

Claim 13 (original): The system of claim 1 where said plural computer applications pertain to at least one of:

(a) stock trading;

(b) communications processing; and

(c) internet services.

Claim 14 (previously presented): The system of claim 1 where at least one of said plurality of objects is copy shared among said plural applications.

Claim 15 (previously presented): The system of claim 1 where at least one of said plurality of objects is direct shared among said plural applications.

Claim 16 (currently amended): A system for concurrent operation of plural computer applications, said system comprising:

(a) a computer storage medium including a shared object space selectively connectable to each of a plurality of computer applications, said shared object space capable of storing:

(i) a plurality of objects accessible to each of said plural computer applications connected to said shared object space; and

(ii) a queue associated with said shared object space and capable of storing a plurality of references to objects, each reference received from one of a first set of said computer applications and identifying an individual object ~~and at least one application from a second set of applications~~; and

(b) at least one computer comprising at least two computer applications concurrently executing on respective virtual machines, a particular object updateable by one of said concurrently executing applications belonging to said second set of said applications when said one application of said second set is connected to said shared object space and in control of an object reference from said queue identifying said particular object, said reference to said particular object received in said queue from one of said computer applications of said first set, said one of said applications of said first set and said one of said applications of said second set not communicating with each other regarding said reference to said particular object.

Claim 17 (original): The system of claim 16 where said queue is a predefined type.

Claim 18 (original): The system of claim 16 where said queue is customized.

Claim 19 (original): The system of claim 16 where said queue is a "first-in-first-out" queue.

Claim 20 (original): The system of claim 16 where said queue is a "last-in-first-out" queue.

Claim 21 (previously presented): The system of claim 16 where each said virtual machine comprises a computer executable instruction in conformance with a virtual machine specification, said instruction executing on said computer.

Claim 22 (previously presented): The system of claim 21 where said shared object space is connected to each said virtual machine through a Native Method Interface.

Claim 23 (original): The system of claim 22 where said system includes a default directory with a native language library file.

Claim 24 (original): The system of claim 16 where said shared object space is operably connectable to a non-object-oriented application.

Claim 25 (currently amended): The system of claim 24 where said non-object oriented application is a "C" program.

Claim 26 (previously presented): The system of claim 16 where access to at least one of said plurality of objects by said plural computer applications is synchronized.

Claim 27 (currently amended): The system of claim 16 wherein said object reference ~~further comprises~~ is associated with a reference to said one computer application of said second set.

Claim 28 (original): The system of claim 16 where said plural computer applications pertain to at least one of:

(a) stock trading;

(b) communications processing; and

(c) internet services.

Claim 29 (previously presented): The system of claim 16 where at least one of said plurality of objects is copy shared among said plural applications.

Claim 30 (original): The system of claim 16 where at least one of said plurality of objects is direct shared among said plural applications.

Claim 31 (currently amended): A system for the concurrent operation of plural computer applications, said system comprising:

(a) a computer storage medium including a shared object space selectively connectable to each of a plurality of computer applications, said shared object space capable of storing:

(i) a plurality of objects accessible to a plurality of said plural computer applications connected to said shared object space; and

(ii) a queue associated with said shared object space and capable of storing a plurality of references to objects, each reference to an object received from one of said plural computer applications and identifying an individual object and an application capable of removing said reference from said queue; and

(b) at least one computer comprising at least two computer applications concurrently executing on respective virtual machines, a particular object updateable by one of said concurrently executing applications when said one application is connected to said shared object space and in control of a reference from said queue identifying said particular object, said one application removing said reference to said particular object from said queue and returning said reference to said particular object to said queue if said ~~reference identifies~~ queue contains an

identity of another application ~~capable of removing~~ seeking to remove said reference from said queue, said one application and said another application not communicating with each other regarding said reference.

Claim 32 (previously presented): The system of claim 31 where said queue is a predefined type.

Claim 33 (previously presented): The system of claim 31 where said queue is customized.

Claim 34 (previously presented): The system of claim 31 where said queue is a "first-in-first-out" queue.

Claim 35 (previously presented): The system of claim 31 where said queue is a "last-in-first-out" queue.

Claim 36 (previously presented): The system of claim 31 where each said virtual machine comprises a computer executable instruction in conformance with a virtual machine specification, said instruction executing on said computer.

Claim 37 (previously presented): The system of claim 36 where said shared object space is connected to each said virtual machine through a Native Method Interface.

Claim 38 (previously presented): The system of claim 37 where said system includes a default directory with a native language library file.

Claim 39 (previously presented): The system of claim 31 where said shared object space is operably connectable to a non-object-oriented application.

Claim 40 (previously presented): The system of claim 39 where said non-object oriented application is a "C" program.

Claim 41 (previously presented): The system of claim 31 where access to at least one of said plurality of objects by said plural computer applications is synchronized.

Claim 42 (currently amended): The system of claim 31 wherein said reference to an object further comprises an indication of a position of said reference in a sequence of references received in said queue from said application placing said reference in said queue.

Claim 43 (previously presented): The system of claim 31 where said plural computer applications pertain to at least one of:

(a) stock trading;

(b) communications processing; and

(c) internet services.

Claim 44 (previously presented): The system of claim 31 at least one of said plurality of objects is copy shared among said plural applications.

Claim 45 (previously presented): The system of claim 31 where at least one of said plurality of objects is direct shared among said plural applications.